5. Klara

Program Name: Klara.java Input File: klara.dat

Klara is learning how to manipulate character grids and decides to do some experimenting with her newly learned techniques. She creates some random square character patterns, anywhere from 4 to 20 as the side dimension of the square, and uses two characters, the pound sign (‘#’) and the dot (‘.’) to fill the square. She then finds the very middle of the square, discovers what that character is, and then proceeds to change every instance of that character in the grid to a dash (‘-‘). However, to make it a bit more of a challenge, she decides that this can only happen if it is “accessible” from the middle, going up, left, down, or right, but not diagonally. If the other character is blocking the way, acting as a wall, she can’t go that way and has to find a way around somehow, if it is even possible. She discovers that sometimes she can’t change all the characters because they are unreachable.

Let’s look at two examples, shown in the boxes below. The size of the first grid is 4, which means the middle character is located in row 2, column 2, which contains a pound sign. All the remaining pounds signs are reachable, except for the one on the top row, which is blocked from being reached by dots, as shown in the second box.

The third box is an eleven by eleven grid, and the middle character of the box is a dot, which means she wants to change all of the dots to dashes. However, she can only reach a few of them as you can see in the second box since the pound signs are blocking the way and there is no way to reach the other dots.

Let’s look at one of her squares, shown in the first box on the right. The side length of the grid is 11, which means it has eleven rows and eleven characters in each row. The middle character of the box is a dot, which means she wants to change all of the dots to dashes. However, she can only reach a few of them as you can see in the fourth box since the pound signs are blocking the way and there is no way to reach the other dots.

```
.#..
..##
..#.
..##
```

```
.#..
..--
..-.
..--
```

```
###..#.#.##
###.#..#.#.
#..##.#.#.#
#.####.#.##
##.##.#.#.#
###.#..#...
.#.##..#.##
###..#..#..
#.####.#.#.
#.....###.#
#.#..##..##
```

```
###..#.#.##
###.#..#.#.
#..##.#.#.#
#.####.#.##
##.##-#.#.#
###.#--#...
.#.##--#.##
###..#--#..
#.####-#.#.
#.....###.#
#.#..##..##
```

Input: The data file will contain several sets of data, each set consisting of an integer N, followed by an N x N square grid of characters as described above.

Output: The resulting grid after changing all possible characters, reachable from the center character, to dashes as described above, followed by the string “=====”.

(Sample input and output on next page)

(Klara continued)

Sample input:

```
4
.#..
..##
..#.
..##
11
###..#.#.##
###.#..#.#.
#..##.#.#.#
#.####.#.##
##.##.#.#.#
###.#..#...
.#.##..#.##
###..#..#..
#.####.#.#.
#.....###.#
#.#..##..##
7
.#.####
#..#...
#...#.#
.####..
#.#...#
##.####
#...##.
```

Sample output:

```
.#..
..--
..-.
..--
=====
###..#.#.##
###.#..#.#.
#..##.#.#.#
#.####.#.##
##.##-#.#.#
###.#--#...
.#.##--#.##
###..#--#..
#.####-#.#.
#.....###.#
#.#..##..##
=====
.#.####
#..#...
#...-.#
.----..
#.-...#
##.####
#...##.
=====
```